Design, Development, and Testing of a Low Cost, Fully Autonomous Indoor Unmanned Aerial System

Girish Chowdhary*, D. Michael Sobers, Jr.†, Erwan Salaün‡, John A. Ottander*, Eric N. Johnson§
Georgia Institute of Technology, Atlanta, GA, 30332-0152, USA

This paper is concerned with the design, development, and autonomous flight testing of the GT Lama indoor Unmanned Aerial System (UAS). The GT Lama is a fully autonomous rotorcraft UAS capable of indoor area exploration. It weighs around 1.3 lbs (600 gms), has a width of about 27.6 inches (70 cm), and costs less than USD 900. The GT Lama employs only five off-the-shelf, extremely low-cost range sensors for navigation. The GT Lama does not rely on other expensive and sophisticated sensors, including inertial measurement units, Laser based range scanners, and GPS. The GT Lama achieves this by using simple wall following logic to ensure that maximum perimeter of an indoor environment is explored in a reasonable amount of time. The GT Lama hardware, and the Guidance, Navigation, and Control (GNC) algorithms used are discussed in detail. The details of a MATLAB based method that facilitates rapid in flight validation of GNC algorithms on real flight hardware is also discussed. Results from flight tests as the GT Lama autonomously explores indoor environments are presented.

I. Introduction

The need for autonomous vehicles has been widely demonstrated for tasks such as search and rescue, disaster assessment, and military reconnaissance, to name a few. Miniature air vehicles are ideal candidates for such missions as they can use three dimensional maneuvers to overcome obstacles that cannot be overcome by ground vehicles. However, significant technological challenges exist in order to ensure reliable operation in such environments. Most current algorithms for Unmanned Aerial System (UAS) Guidance Navigation and Control rely heavily on GPS signals,[1–3] and hence are not suitable for indoor navigation where GPS signal is normally not available. Furthermore, the indoor UAS must be sufficiently small in order to successfully navigate cluttered indoor environments, consequently limiting the amount of computational and sensory power that can be carried onboard. Finally, the UAS may need to be expendable due to the dangerous environments it needs to operate in, hence low-cost, lightweight designs need to be explored. These restrictions pose significant technological challenges for the design of reliable Miniature Air Vehicle (MAV) platforms capable of navigating cluttered areas in a GPS denied environment.

In order to allow for sufficient manoeuvrability, researchers have relied on rotorcraft based MAVs. Many rotorcraft based configurations, including Quadrotor MAVs, ducted fan MAVs, and conventional single rotor based MAVs are inherently unstable. To counter this lack of inherent stability, researchers have proposed the use of strap-down Micro-Electro-Mechanical Systems (MEMS) inertial sensors (gyroscopes and accelerometers) to aid in the attitude stabilization of intrinsically unstable platforms (e.g. see Refs. 4, 5 for Quadrotor stabilization). In order to bound the biases from inertial sensors other sensors providing position-related information are also employed for indoor navigation: e.g. Laser rangefinder,[6] vision system,[7,8] or both.[9] With all the required embedded sensors to enable stable flight of inherently unstable MAV platforms, the avionics system necessary for vehicle stabilization and indoor navigation ends up being heavy

*Graduate Research Assistant, School of Aerospace Engineering, AIAA Student Member

†Major, USAF, PhD; Recent Graduate, School of Aerospace Engineering, AIAA Member. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

‡Postdoctoral Fellow, School of Aerospace Engineering, AIAA member

§Lockheed Martin Professor of Avionics Integration, School of Aerospace Engineering, AIAA Member

and expensive. On the other hand, if a rotorcraft platform that is stable in the attitude loop is used, then the required sensor load could be significantly reduced, resulting in significant decrease in the cost and complexity of the system. It is the purpose of this paper to establish the feasibility of a low-cost, low-weight approach based on inherently stable MAV platforms.

In this paper we present the details of the design, development, and flight testing of the GT Lama UAS which is capable of navigating cluttered and confined environments with low-cost and low weight embedded avionics system. The GT Lama has been developed by the Georgia Institute of Technology's Unmanned Aerial Vehicle Research Facility (UAVRF). It weighs around 600g, is 70cm wide, and costs less than USD 900. The GT Lama employs a novel approach to indoor Guidance Navigation and Control (GNC) that does not require inertial sensors, vision system, or GPS signals, in fact it relies *only* on five commercially available, low-cost range sensors and an event based guidance algorithm that uses the internal geometry of indoor environments and leverages the inherent stability properties of the coaxial rotorcraft platform used to ensure that maximum amount of indoor area is explored in a reasonable amount of time. During the development phase, the GT Lama was used as Georgia Institute of Technology's primary entry for the 2009 AUVSI International Aerial Robotics Competition.[10,11] The GT Lama took second place in the competition. Along with the details of the GT Lama design, Guidance, Navigation, and Control algorithms, we also describe a novel method for rapid validation of GNC algorithms that employs off the shelf hardware, and the commercially available MATLAB software environment. This method allows the control designer to move the bulk of GNC computation off-board to the user friendly MATLAB operating environment using off-the-shelf datalinks. This allows rapid development of control algorithms, real-time plotting, and post-flight data analysis, expediting greatly the process of in-flight validation of GNC concepts on flight hardware.

We begin by discussing the miniature rotorcraft platform employed for the development of the GT Lama. We then discuss the details of the hardware integration process, including describing the sensing, computational, and communications hardware used. We then provide a brief overview of the event driven guidance, navigation, and control policy employed on the GT Lama. We then describe the MATLAB architecture used for rapid in-flight validation of GNC algorithms and sensor hardware. The details of the onboard embedded implementation of the algorithm are then presented. Finally we present flight test results as the GT Lama autonomously explores indoor environment.

II. Vehicle and embedded avionics system

A. Aerial Platform

A vehicle designed to traverse indoor environments must be sufficiently small and be able to maneuver tight corners. A traditional fixed-wing platform is unable to hover, and hence is severely disadvantaged in this case. A natural choice then would be rotorcraft based designs. However, most rotorcraft are unstable in flight and must have an inertial measurement unit (IMU) along with additional algorithms and sensors to estimate velocity, attitude, and angular rates in order to be effectively stabilized using closed loop control laws. Due to restrictions on size and weight, strap-down MEMS based IMU must be utilized. The data from these sensors is susceptible to sensor drift and misalignment errors which must be corrected for using an external position fix.[1] Typically, this is achieved by using GPS signals,[1–3] which are unavailable in indoor environments. A solution to this problem is to circumvent the attitude stabilization task by choosing a vehicle that has desirable stick free stability properties in attitude. An example of such a vehicle is a coaxial miniature rotorcraft platform with counter-rotating blades and a Bell stabilizer bar.

The coaxial miniature rotorcraft platform selected for this research is the E020 "Big Lama"[a], made by E-Sky® (see figure 1). The vehicle is a counter-rotating coaxial helicopter with no tail rotor. The upper rotor is stabilized by a Bell stabilizer, while the lower rotor is connected to a 2-servo swash plate. The system is a four channel helicopter with pitch, roll, yaw, and throttle control, with a yaw-damping gyro to improve handling qualities. Some additional technical specifications for the stock aircraft are: main rotor diameter of 460 mm, takeoff weight 410 g, 800 mAh 11.1 V LiPo Battery, 75 MHz FM radio. The counter rotating blades of this vehicle ensure that the net torque on the airframe is nearly eliminated. The vehicle was augmented with off-the-shelf Heading Hold Gyro which ensures that the vehicle maintains its attitude (rotational angle around the body z axis). The Bell stabilizer bar provides this vehicle with a passive stabilization system. The effect of the Bell stabilizer bar can be briefly explained as follows: the Bell stabilizer bar essentially

[a] http://www.twf-sz.com/english/products.asp?prodid=0291

Figure 1: The Big Lama. Photo courtesy E-Sky®. *Note: the tail rotor on this aircraft is neither functional nor required.*

consists of a decoupled gyroscopic element that tends to hold its attitude in space. If the vehicle encounters a disturbance that changes the attitude of the airframe, the Bell stabilizer remains fixed in space and the resulting attitude difference between the main part of the airframe and the Bell stabilizer causes a restoring effect on the airframe.

B. Flight Avionics

With a stable attitude loop, the main task of the onboard controller is to provide position control by directly linking servo deflections to position measurements. These position measurements constitute of relative altitude and relative distance to nearby features of the indoor environment. Considering restrictions imposed by indoor flight, the driving factor in sensor selection were maximum possible resolution and range with minimum possible weight, size, cost, and power. The MaxBotix® LV-MaxSonar® [b] (shown in Figure 2a), and the SHARP GP2Y0A02YK0F[c] (shown in Figure 2b) were chosen. The Maxbotix sonar has a wide beam width, and returns orthogonal range measurements within a 30 degree tilt range, hence this sensor was chosen for altitude measurement. The Sharp infrared sensors have a narrow beam width and can be used to precisely measure point distances. Hence these sensors were selected for measuring range to walls and obstacles. See Table 1 for detailed information on the sensors. The final configuration of sensors is shown in Figure 4. The layout consists of one sonar range sensor pointed downwards to measure the altitude, two IR range sensors installed with a 45 cm offset to measure distance to obstacles in front of the rotorcraft, and two IR sensors arranged to measure the distance to obstacles on the right and left of the aircraft. Difference in range measurement between the two forward looking IR sensors can also be used to estimate the heading of the aircraft. Details of the sensor properties can be found in Reference 12.

Table 1: Range Sensor Manufacturer Specifications and Price Ranges

Sensor	MaxSonar®	SHARP IR
Range	0.15-6.45 m	0.2-1.5 m
Resolution	2.54 cm	1 cm
Weight	4.3 g	4.8 g
Price	∼ USD 25	∼ USD 13

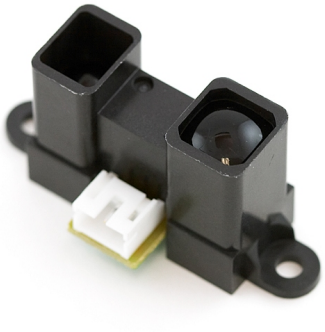
An ATmega128 onboard microcontroller[d] was used as the flight computer for GNC software. All software was programmed in C using the AVR Studio software development environment and uploaded onto the

[b] LV-MaxSonar®-EZ4™ Data Sheet, MaxBotix® Inc., 2007.

[c] SHARP GP2Y0A02YK0F Distance Measuring Sensor Unit Data Sheet, Sheet No.: E4-A00101EN, SHARP Corporation, 2006.

[d] ATMEL® ATmega128©with AVR-H128C Header board with 10 pin ICSP connector, 4KB SRAM, datasheet OLIMEX

(a) MaxBotix® LV-MaxSonar®



(b) SHARP GP2Y0A02YK0F Infrared Sensor

Figure 2: Lightweight, low-cost range sensors suitable for indoor navigation. Photos credit Sparkfun™ Electronics.

Figure 3: Lightweight, low-cost 2.4 GHz XBee modem. Photo credit Sparkfun™ Electronics.

ATmega using a JTAG connector. We found the ATmega 128 processor with the 16 MHz clock speed, and the associated support documentation and software satisfactory for the purpose of this effort. The IR range sensors have analog voltage outputs, which was read by onboard analog to digital input channels. The altitude sonar was read via serial port. The data was transmitted to the ground using the commercially available 2.4 GHz XBee® XBP24-AWI-001 modem (shown in figure 3) employing the IEEE 802.15.4 standard. The XBee weighs less than 5 g, costs around USD 40, and has a range of about 90 m (300 ft) in indoor environments[e]. The final flight configuration, with the platform, avionics, a larger battery, brushless motors, and a protective shroud weighed only 605 g with a rotor diameter of less than 1 meter and had a total cost of less than USD 900 (valued on August 2008). The final flight configuration is shown in figure 6.

III. Navigation Algorithm

A key focus of this effort was to enable navigation in indoor environments without relying on any external positioning signals. As mentioned in the section on vehicle selection (Section II.A), the baseline platform was chosen for its desirable stick free attitude stability properties. This property was leveraged to minimize the weight and cost of the vehicle by employing a bare minimum sensor configuration. The configuration included only 5 range sensors, details of which are given in the section on flight avionics (Section II.B).

Let h denote the altitude measurement, let z denote the distance to an obstacle in front of the rotorcraft,

Ltd. 2006

[e]Digi XBee 2.4 GHz XBee XBP24-AWI-001 60 mW Wire Antenna Datasheet, Digi Inc. (formally Maxstream) 2008.

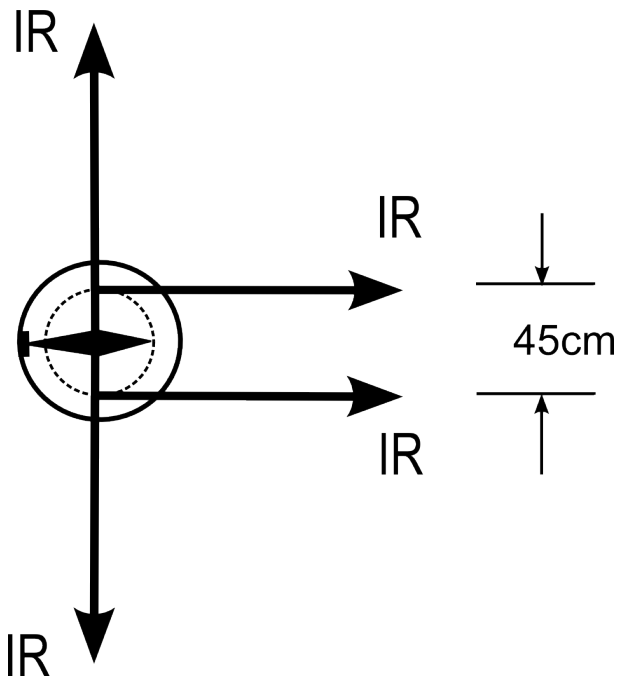


Figure 4: Range sensor layout.

and let y_R and y_L denote the distance to obstacles on the right and left of the rotorcraft. Then the state vector to be estimated is $z = [h, x, y_R, y_L]$. The information from these sensors is subject to noise and interference from companion sensors, and must be further processed. Furthermore, the control system for this vehicle uses a Proportional Derivative Integral (PID) control architecture to directly control desired relative position of the aircraft. Hence, estimate of the time differential of relative position $\dot{z}$ (relative velocity) is required. In order to avoid numerical issues brought about by numeric differentiation and to provide an architecture for smoothing outliers, a Kalman Filter based smart outlier detection and velocity estimation algorithm was used. Let z, be the state of the system and $\dot{z}$ be its first derivative, then following system describes the dynamics used for estimating $\dot{z}$:

$$\begin{pmatrix} \dot{z} \\ \ddot{z} \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} z \\ \dot{z} \end{pmatrix} \tag{1}$$

Suppose z is available as sensor measurement, then an observer in the framework of a Kalman filter can be designed for estimating $\dot{z}$ from available noisy measurements using the above system.[1,13] Let ϕ be the state transition matrix, R be the measurement covariance matrix, Q be the process covariance, P be the state covariance, H be the observation matrix, and $\hat{z}$ is the state estimate. Choosing appropriate constant R and Q, a steady state covariance P can be found by solving the Discrete Algebraic Riccati Equation (DARE). The Kalman gain K can then be computed as follows:

$$K = PH^T(HPH^T + R)^{-1} \tag{2}$$

When a measurement is processed the estimated state is first propagated forward to the current time:

$$\hat{z}^- = \Phi\hat{z} \tag{3}$$

Then the estimated state is updated using the measurement y:

$$\hat{z} = \hat{z}^- + K(y - H\hat{z}^-) \tag{4}$$

A traditional Kalman filter is ineffective in handling discontinuities and bad measurements from the sensors as these cannot be modeled as zero mean Gaussian white noise. A bad measurement is defined as a measurement between the minimum and maximum valid outputs of the sensor which appears to be uncorrelated with the

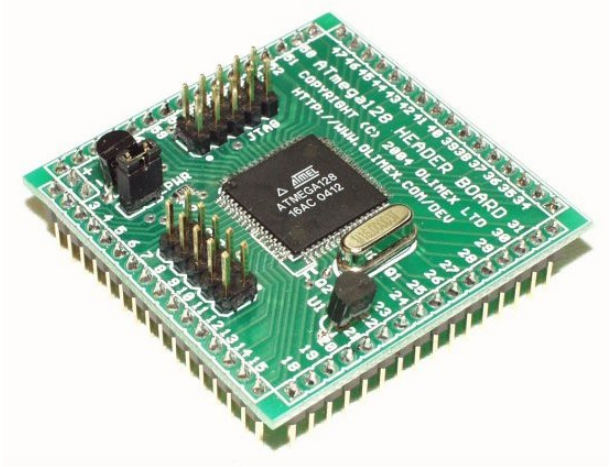


Figure 5: The Atmel ATmega 128 onboard microcontroller serving as the onboard computer for GNC software, photo credit, Atmel Inc.

previous and the latter measurement. When an update using a bad measurement occurs the resulting large residual between the predicted measurement and the actual measurement will significantly change the state and its derivative; affecting the controller performance adversely. In addition, if an obstacle suddenly comes into range of a sensor, a non-existent velocity will be predicted, again adversely affecting the controller performance. To overcome the problems brought about due to bad measurements and "pop-up" obstacles, the Kalman filter was augmented with an outlier detection filter, as shown in Figure 7. The outlier detection filter checks whether the residual for every new measurement is statistically probable (i.e. less than 3 standard deviations away from the residual covariance). If this is indeed the case then the measurement is accepted and an update is done using equation 4. The covariance of the residual can be approximated by:[13]

$$H(\Phi P \Phi^T + Q)H^T + R \tag{5}$$

If the residual is deemed improbable the measurement is rejected but stored and the state is propagated without a measurement update. If the measurement is rejected 4 times in a row and all the rejected measurements are sufficiently similar, then a discontinuity (possibly due to an obstacle or change in the indoor geometry) is detected and the new measurement is accepted without altering the velocity estimate, this ensures smooth control. Figure 8 shows an example of the filter in operation.

The heading of the rotorcraft is detected using the relative range difference between the two forward mounted IR sensors. Let the measurements from these sensors be denoted as x_R and x_L for the right and the left sensor respectively and let L denote the mutual horizontal offset between the two sensors. Then an estimate of the relative heading ψ can be given as:

$$\psi = \arctan \frac{x_R - x_L}{L}. \tag{6}$$

Figure 9 depicts the equations and the associated quantities used for detecting the heading and distance of the aircraft relative to a wall. IR sensors are accurate only within a limited range of an obstacle, hence this heading estimate is highly susceptible to noise. This issue is handled in a computationally efficient manner by filtering the IR range measurements using an Extended Kalman Filter.[13] The orthogonal distance to the obstacle in front, denoted previously by x can then be found by averaging x_R and x_L and correcting for the heading ψ. Figure 10 shows the performance of the navigation algorithm attempting to estimate the wall distance and relative heading with respect to the wall using two offsetted IR range sensors.

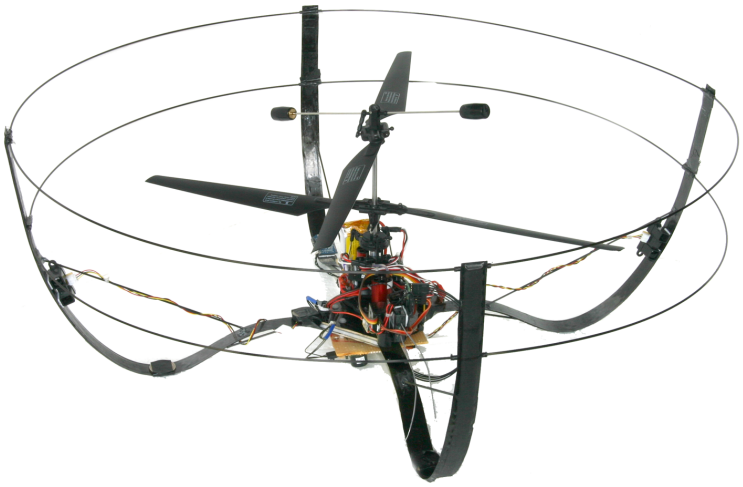


Figure 6: Flight vehicle with sensors and protective shroud installed.

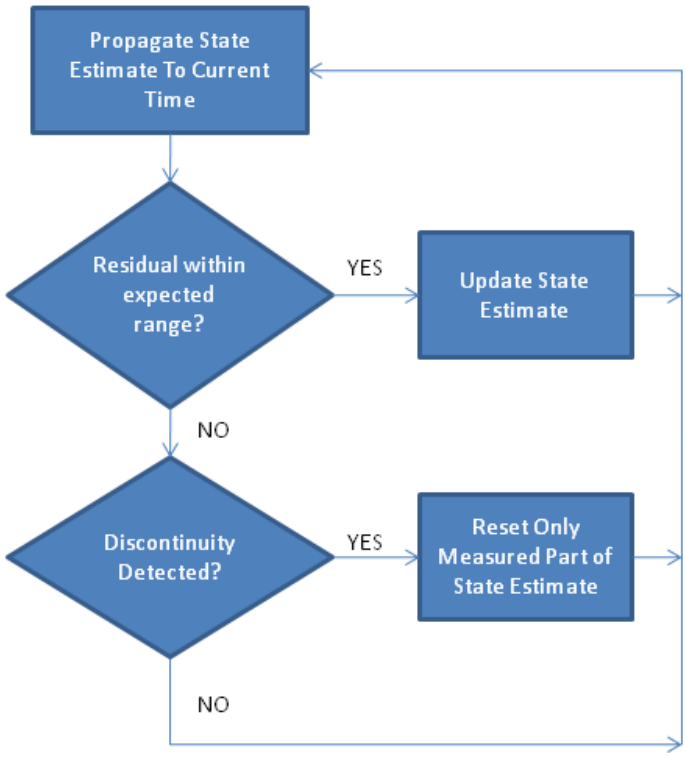


Figure 7: Flow chart of outlier detector used to reject bad measurements and detect discontinuities.

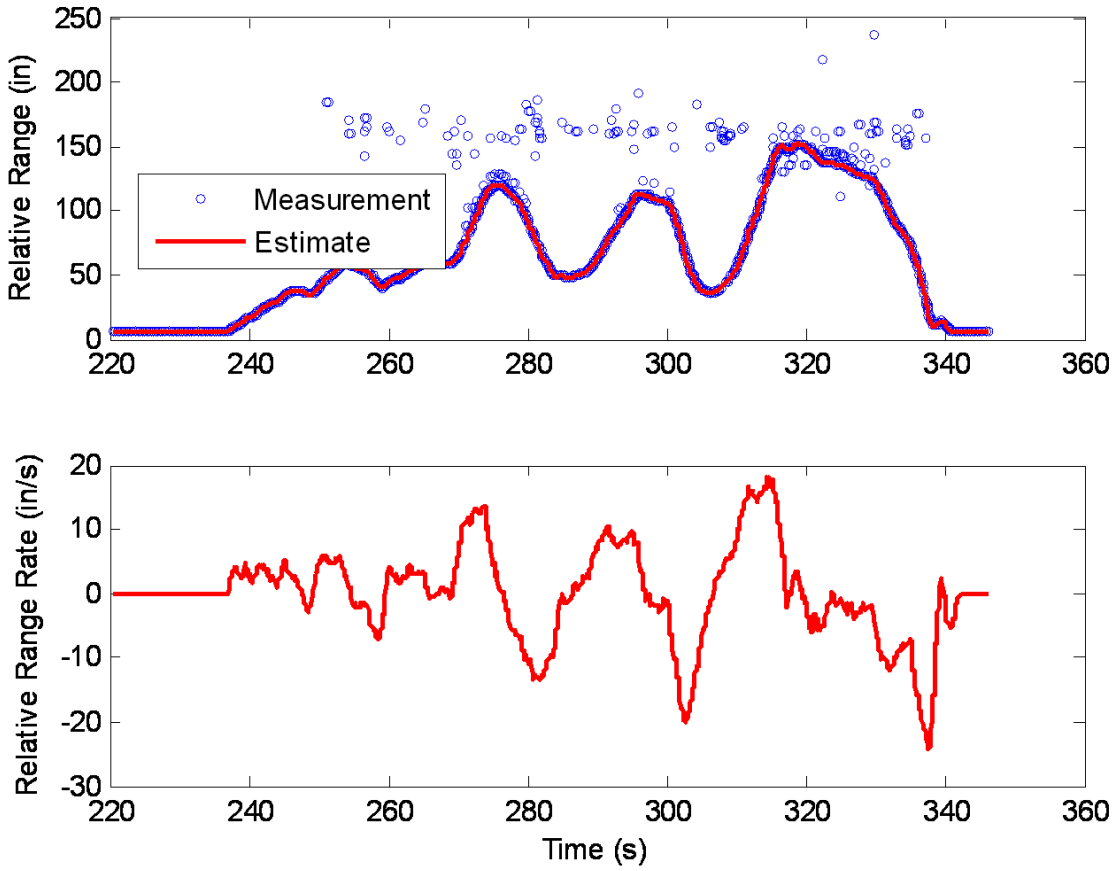


Figure 8: Sonar altitude measurements recorded during flight. Note that the smart filtering routine is successfully able to maintain the range estimate in the presence of outliers.[10]

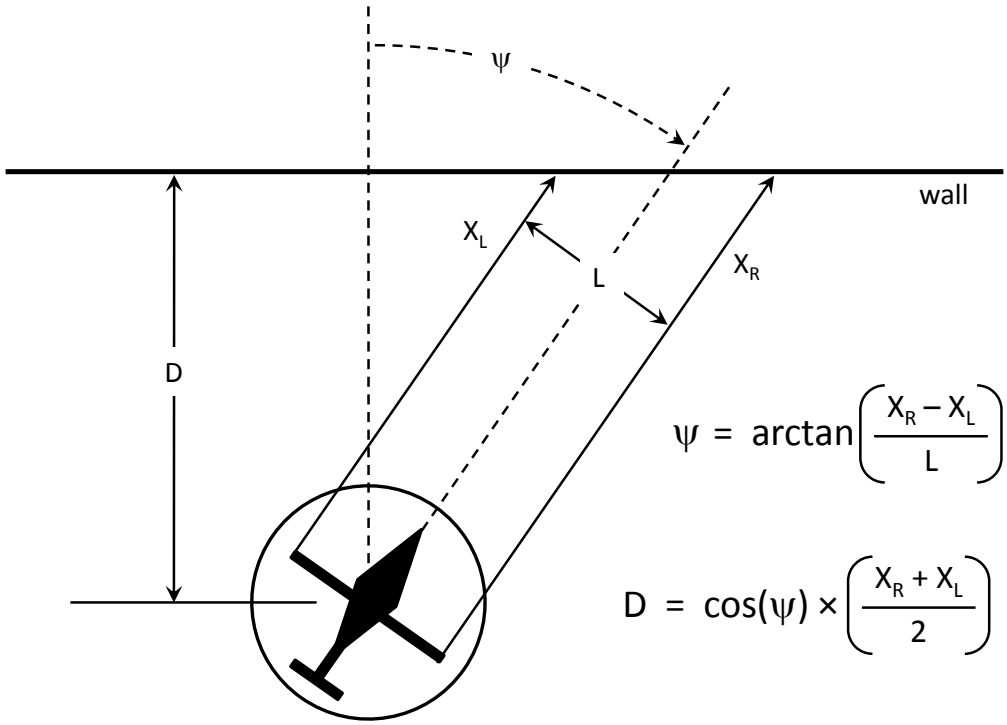


Figure 9: The two forward-looking IR sensors are used to calculate vehicle heading with respect to the wall (ψ). Relative heading and average range on the two sensors is used to calculate perpendicular distance to the wall (D).

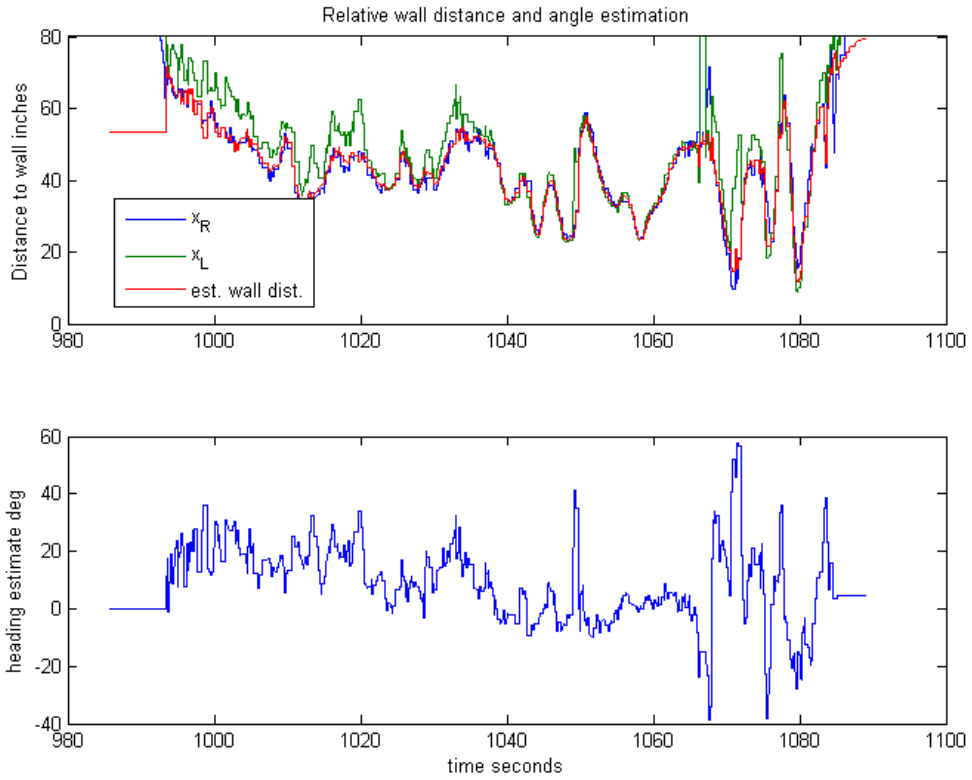


Figure 10: Relative wall distance and relative heading w.r.t wall estimation using two offsetted IR sensors

IV. Guidance Algorithm

The task of the Guidance algorithm is to plan a path such that the mission objectives are met. We assume that the mission goal is to relay live video information from indoor area of interest to a remote observer. Traditionally, the path of planning an optimum path such that mission requirements are satisfied is solved by using predefined global maps and the knowledge of the absolute information of the vehicle with respect to the map.[14] This approach however, cannot be used in our case as no absolute position information or a predefined map of the environment is available. MAVs operating in indoor environments need to simultaneously gather information about the immediate surrounding (also known as mapping) and find its position within this surrounding (also known as localization).

Various approaches have been used towards solving the Simultaneous Localization and Mapping (SLAM) problem, with a common one being the integration of vision sensors with inertial sensors.[15–17] This approach relies on sophisticated measurements of the environment and needs significant computational resources for simultaneously forming a map and estimating the position of the vehicle. Since the main goal of our work was to reduce the cost and the weight of the aircraft as much as possible, we used an alternate approach that allows indoor area exploration without having to first solve the SLAM problem. Our approach is based on the simple fact that all indoor environments have walls. Walls are easy to identify, have a smooth structure, and are built in a predictable manner. Furthermore, it is possible to traverse the complete perimeter of an indoor environment by simply detecting a wall and traversing alongside. Consequently, the guidance algorithm is designed to find a wall on its right, left, or in front. After detection, the algorithm points the rotorcraft towards the wall and attempts to maintains a predefined distance from the wall using forward looking range sensor. Measurements from two forward looking range sensors are combined to obtain a relative heading with respect to the wall. This information is then used in the heading controller that ensures that the rotorcraft maintains a constant heading with respect to the wall. In order to traverse the indoor environment, the vehicle simply moves sidewards until it detects an obstacle or another wall in the direction of travel, this triggers a corner guidance algorithm. The detected corners are maneuvered through open loop turn logic.

On the onboard controller, the guidance algorithm is implemented as an event-based algorithm, that is the algorithm switches through different guidance modes based on detection of events. The vehicle starts by entering the indoor arena through a specifically designed "Window Entry" mode. In this mode, an object detected on the forward-looking left or right IR sensors causes the lateral controller to adjust the flight path to remain centered on the window. Once the vehicle enters the indoor arena and walls are detected by the left or right IR sensor, the vehicle switches into "Left Turn" or "Right Turn" mode accordingly. In these mode, an open-loop left or right turn is commanded until the forward-looking IR sensors detect the wall. A confirmed detection of wall switches the guidance mode to "Wall Follow" mode. In this mode, the longitudinal controller maintains a predefined distance from the wall, while the heading controller maintains the desired heading with respect to the wall. The vehicle then uses a predefined open loop lateral cyclic servo deflection to traverse along the side of the wall to the right, using the right facing sensor to detect walls and obstacles in the flight path. If an obstacle or wall is detected by the right looking IR sensor, different corner-turning modes are initiated. If a wall or obstacle is detected in the direction of flight, the vehicle enters "Inside Turn" mode, which is designed to either maneuver the corner (for concave corners) or fly around the obstacle. This is achieved by enforcing a predefined open-loop yaw command until no obstacle is seen by the right IR sensor. If one of the forward-looking IR sensors detects a step change to max range while the other sensor still reads near the estimated wall distance, a convex, or outside, corners has been detected and the vehicle enters the "Outside Turn" mode. In this mode it yaws to the left in order to continue around the corner. At the conclusion of the inside or outside corner mode the vehicle should be facing a wall, this is confirmed by verifying acceptable range measurements from the two front IR sensors. The vehicle then returns to "Wall Follow" mode. and continues flight. In this way it is possible to traverse significant indoor areas in reasonable amount of time. This logic was tested onboard the GT Lama in different indoor arenas and was demonstrated to ensure safe and efficient indoor flight without reliance on GPS signal, inertial measurements, and using only 5 range sensors. Figure 11 presents a schematic of the guidance algorithm.

The above event based guidance algorithm uses various flags (such as "wall detected", "wall on right" etc.) to switch between different control modes. In order to ensure smooth switching, the flag values were filtered using a first order filter, this ensures that a particular mode will be triggered only if the triggering flag has been consistently on for a while, effectively ensuring high confidence in the switches and reducing the possibility of chance switches.

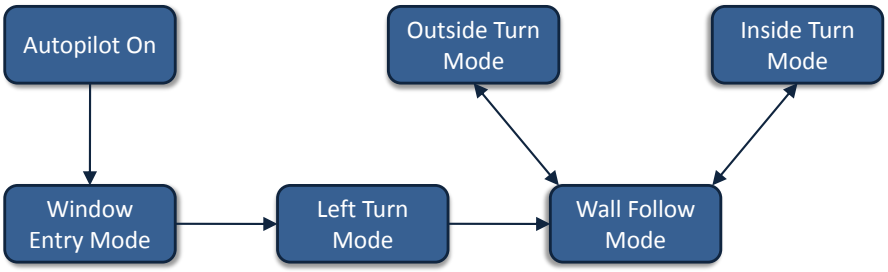
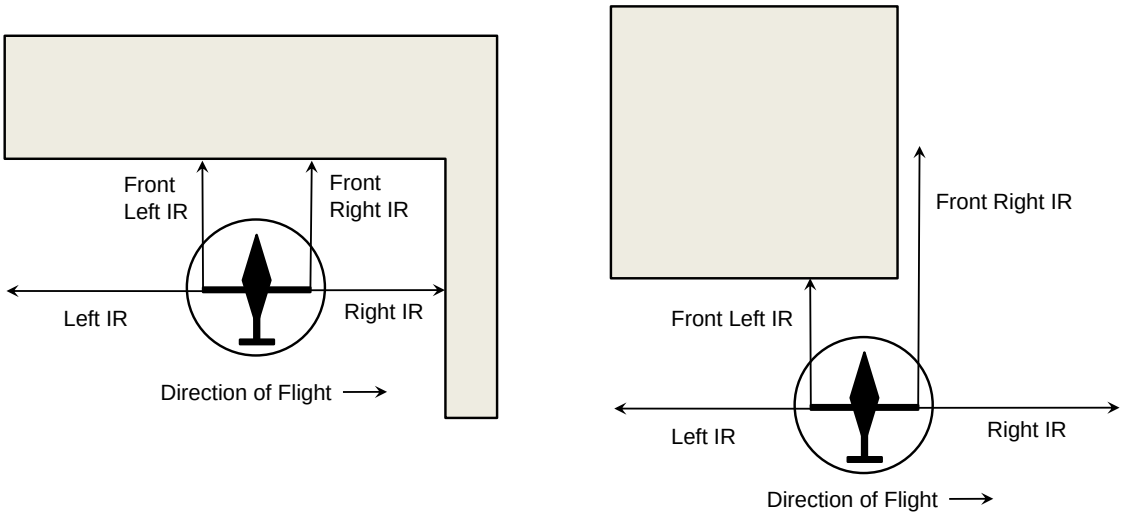


Figure 11: Wall-following guidance algorithm. Specific sensor inputs will cause the algorithm to progress to successive logic blocks.



(a) Inside turn

(b) Outside turn

Figure 12: When the vehicle is flying laterally, sensing a wall in the direction of flight triggers the guidance state to enter an inside turn as show in (a). If a the forward-looking range sensors detect an outside corner as shown in (b), the guidance system state switches to outside turn mode.

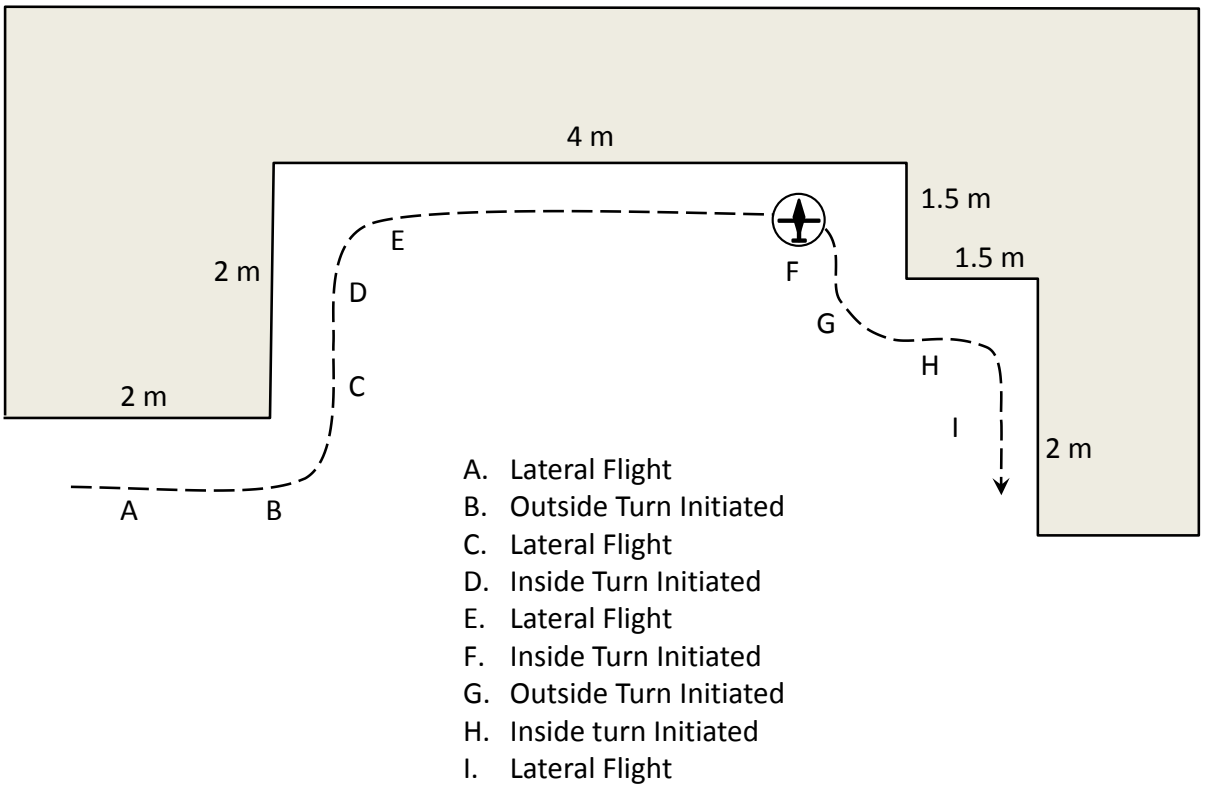


Figure 13: A typical flight test environment. In this test, the vehicle begins at the left side of the room and flies to the right while maintaining a specified distance from the wall. When outside and inside turns are encountered, the vehicle guidance system switches state and performs the appropriate behavior.

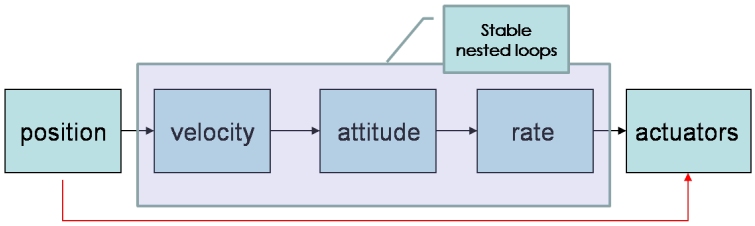


Figure 14: position commands are linked directly to servo deflections by leveraging vehicle stick free attitude stability of properties (which result in the stable nested loop shown in the figure). If the nested loop were not stable, as is the case for conventional rotorcraft or quad-rotors, a traditional method requiring active rate and attitude hold needs to be employed

V. Control Algorithm

In the traditional method for rotorcraft control, a position loop commands velocity, the velocity loop commands attitude, and the attitude loop achieves stabilization by using the actuators to control the angular rate. This approach is known as nested control loop stabilization and has been widely studied.[14,18–20] For the GT Lama, due to the lack of accurate angular rate and attitude information, the traditional method of nested control loops cannot be utilized. We approach this problem by leveraging the inherent stick free stability properties (in roll and pitch) of the coaxial rotorcraft platform and an off-the shelf heading lock gyro to control yaw. The position command is then directly linked to the actuator deflection using a Proportional Integral Derivative (PID) control for providing servo deflections such that the vehicle is able to track a commanded relative position. The position commands are designed to ensure that the vehicle maintains a predefined relative distance from a wall and maintains relative altitude. Figure 14 shows a schematic of our approach. The control architecture used is a PID design with gain scheduling applied such that the controller uses different gain values depending on where the vehicle is with respect to the wall and the direction of its relative velocity with respect to the wall.

The rotorcraft dynamics around a trim point (such as hover) can be approximated by a linear model of the form [19,21,22]

$$\dot{x} = Ax + B(u + u_{trim}) \tag{7}$$

where x is the estimated state of the system including the position, velocity, and the angular rates. The control input is given by u. Note that this control input is intended to provide a correction around the trim of the rotorcraft, given by u_{trim}. We discuss the design of the altitude PID control loop. Let the estimated relative altitude be given by h and the commanded altitude be given by h_c which can be a constant command or the output of a reference model chosen to characterize the response of the system:

$$\dot{h}_c = A_{rm}h_c + B_{rm}r \tag{8}$$

where $r(t)$ is a reference input to the reference model. The error between the estimated state and the command is:

$$e = h - h_c \tag{9}$$

This results in the error system:

$$\dot{e} = Ae + (A - A_{rm})z_c + Bu - B_{rm}r \tag{10}$$

The PID control action can be summarized by the following equation:

$$u = K_p e + K_d \dot{e} + K_i \int_{t_0}^{t} u(t)dt \tag{11}$$

In the above equation, instead of integrating the error $e(t)$ as is traditionally done, the servo commands output by the controller is integrated (under the assumption that the servo output is linearly dependent on the error). In this way the system can inherently handle actuator saturation and integration windup.

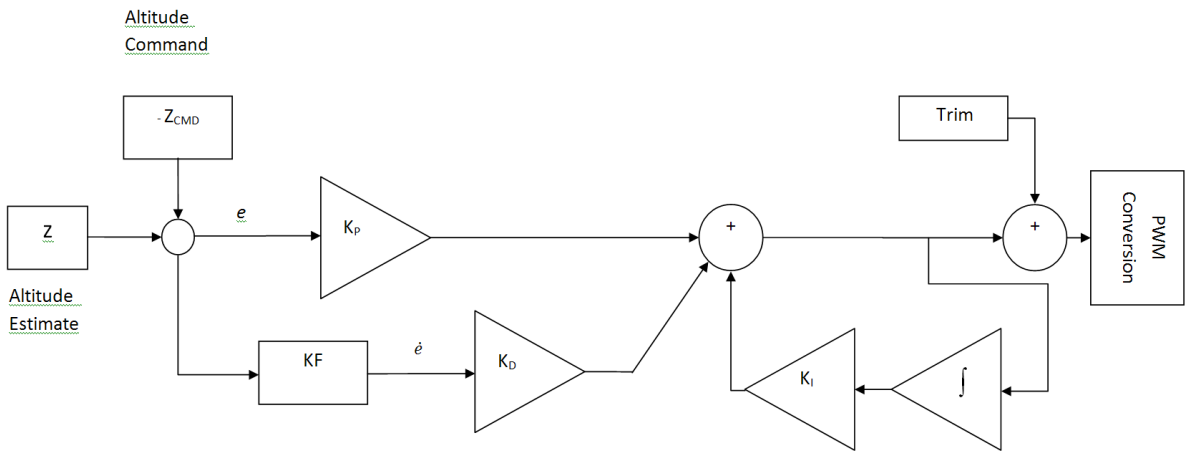


Figure 15: Architecture for the altitude hold controller, KF denotes the Kalman Filter.

Furthermore, servo commands are easier to measure since they are assigned by the controller. Closed loop stability is ensured by choosing the PID gains $K_p \in \Re$,$K_d \in \Re$,$K_i \in \Re$ such that the error system (Equation 10) is rendered stable through negative feedback. The key to successful and reliable implementation of PID control lies in the estimation of the state, and its derivative. These are estimated using Kalman filter based algorithms as described in the section on navigation (section III). The control action is achieved by using three independent control loops:

Altitude Hold: The function of the altitude hold control loop is to use the filtered measurements from the downward pointed sonar for altitude control. A PID architecture is used, where the derivative of the position is calculated using the Kalman filter architecture as described in section III. During vehicle operation, varying battery voltage level affects the throttle trim value. To counter this effect, an integral part is required in the controller. Figure 15 shows the schematic of the altitude control loop. The lateral and the longitudinal control loops have a similar architecture.

Heading Control: The function of this control loop is to control the heading of the vehicle. The heading information is formulated by using dual IR range sensors mounted with a mutual offset, as described in section III. The purpose of the heading hold is to maintain a relative heading with respect to a local reference, such as a wall. In Inside Turn and Outside Turn modes as described in section IV, the heading control uses open loop commands for rotating around the Z axis.

Longitudinal Position Control: The function of the longitudinal control loop is to ensure that the vehicle maintains a relative distance from a wall or obstacle in front of the vehicle to avoid collision and to ensure that the IR range sensors are in range. Longitudinal control is achieved by using forward mounted IR measurements. The operation of the vehicle in the vicinity of a wall presents a unique challenge for tuning the PID gains. It was observed that the vehicle dynamics are dependent on the distance of the vehicle from the wall, possibly due to interaction of rotor induced flow with the wall structure. It was observed that if the vehicle is within a range of about 0.6 m (2 feet) of the wall, the airflow tends to pull the vehicle towards the wall. Whereas if the vehicle is farther than about 0.9 m (3 feet) from the wall, the airflow tends to push the vehicle away from the wall. To counter this effect, a gain scheduled controller was designed that uses high gains only if the vehicle is closer than 0.6 m (2 feet) from the wall and if the vehicle is approaching the wall. In all other quadrants of the error space the controller interpolates between high and low gains dependent on the distance from the wall. A schematic showing the area of the error phase space where high gains are used is shown in Figure 16. At a range greater than 1.52 m (60 inches), the IR range measurements cannot be fully trusted, hence if the distance of the vehicle is further than 1.52 m (60 inches) from the wall, the vehicle enters an openloop feedforward command forward in vehicle coordinates until the front IR range sensors are back in range.

Lateral Position Control: Lateral position control ensures that the vehicle does not collide with an obstacle in it's lateral path. This is achieved by using similar logic as the longitudinal position control loop.

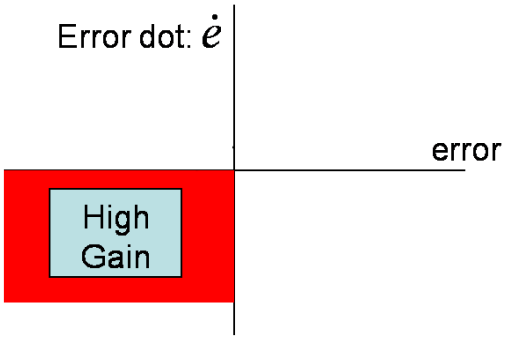


Figure 16: Quadrant of the error phase space where high gain is used

VI. A Method for Rapid In-flight Validation of GNC Algorithms

Indoor vehicles are required to remain small and low cost as they must operate in confined, cluttered, uncertain environments and may be considered expendable. These stringent requirements put considerable constraints on the available computational power onboard the vehicle. Due to this reason, researchers often use highly integrated embedded onboard computers for their processing needs. On the other hand, the algorithms required to successfully navigate cluttered indoor arenas have a tendency to become highly complicated due to the severe uncertainties involved in indoor navigation guidance and control, often handled through an event based structure. Hence, the implementation of these algorithms on onboard processors requires significant efforts in embedded programming and an infrastructure for debugging, validating, and flight testing. Due to these reasons, it is seen that the research in indoor systems becomes easily separated in two phases. The first phase being the development of algorithms, software, and methods that are expected to improve the capabilities. These algorithms are often developed in MATLAB based environment. The second phase is the actual flight implementation and validation of the proposed algorithms for demonstrating real-time feasibility of the algorithms onboard indoor systems. This phase requires significant efforts in converting the MATLAB based algorithms to a form that is conducive to embedded implementation (often using C, C++, or other highly structured programming languages), sensor integration, datalink development, and in-flight validation.

To facilitate the transition from phase 1 to phase 2, researchers have used the VICON system for testing and developing indoor GNC algorithms, which consists of several cameras and associated software that use markers on vehicles to estimate their position and attitude. For example Valenti,[23] How,[24] and Saad[25] have used the VICON system for rapid prototyping of indoor GNC algorithms on indoor vehicles. In this paper we present another approach for transitioning between phase 1 and phase 2 which relies on combining the inherent tools in MATLAB for serial communication along with real flight hardware. The implementation of this approach is as follows:

1. Install sensors onboard the flight vehicle
2. Read sensor data using simple drivers implemented on a low cost low power onboard processor (for example the ATMEL ATmega 128)
3. Transmit the sensor information using off-the-shelf datalink such as the XBee (described in Section II.B) to a desktop computer
4. Read in the sensor information to MATLAB using MATLAB's inherent serial communication tools
5. Use MATLAB scripts to perform required Guidance, Navigation, and Control computations and generate the required Pulse Width Modulated (PWM) signals
6. Transmit the PWM signals to the rotorcraft by connecting to the trainer port on the vehicle Remote Control (RC) transmitter using off-the-shelf servo controllers such as Tom's RC Servo Controller SC-8000[f].

[f]Tom's RC Servo controller SC -8000, Tom's RC inc. http://www.tti-us.com/rc/index.htm (Jan 2010)

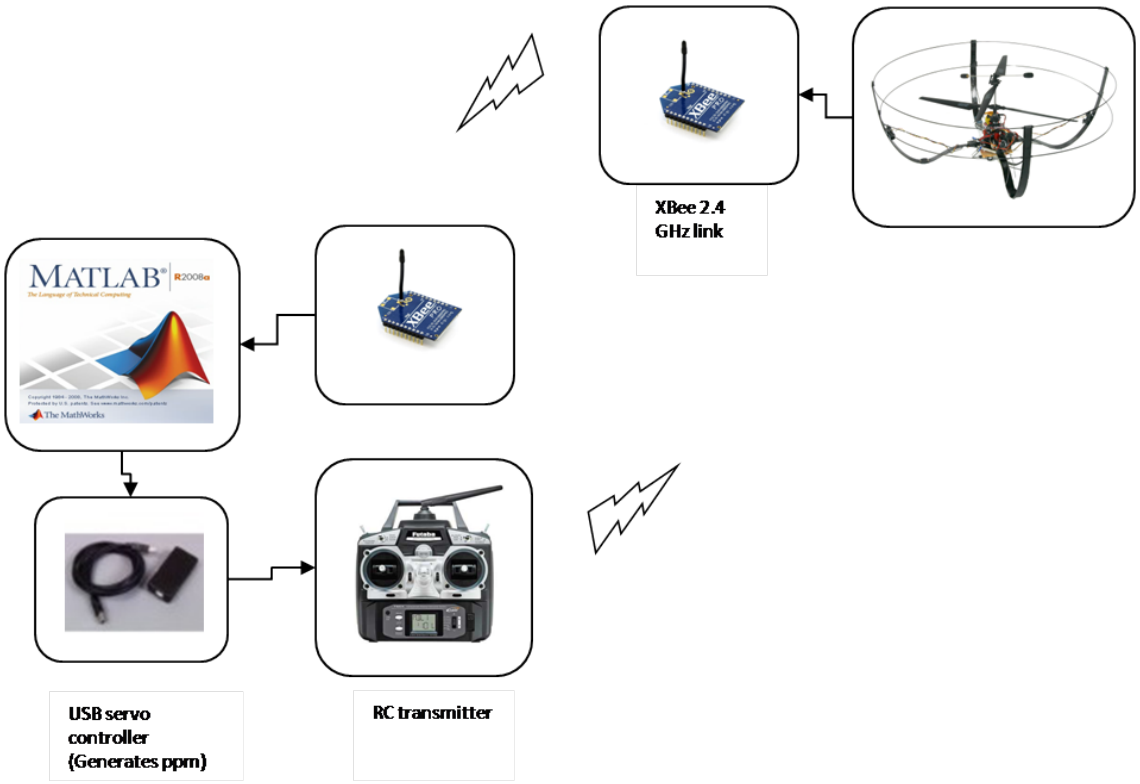


Figure 17: Schematic of the novel architecture used to rapidly test GNC algorithms on real flight hardware (MATLAB® image credit MATHWORKS Inc., Tom's RC USB controller image credit Tom's Robotics Inc., RC transmitter image credit FUTABA Inc.), XBee image credit Sparkfun™ Electronics.

This approach allows the control designer to leverage a host of inbuilt MATLAB functions as well as allows real time plotting, expediting greatly the process of validating GNC concepts in real-time on flight hardware. Furthermore, this method renders the architecture highly flexible, since all servo manipulations are done through RC transmitters exploiting the in built RC receiver of the airframe. This approach is designed to encourage hardware based validation of GNC algorithms early on in the design phase. After algorithms have been validated, the transition to phase 2 can be accomplished by translating the MATLAB based algorithms to an onboard implementation.

The MATLAB part of this scrip relies on the use of MATLAB low level file input output functions[g]. The process begins with creating a serial port using the command *serial*, the port can then be opened using $fopen$ and data can be written and read using $fwrite$ and $fread$ functions. We used C structures stored in text files for transmitting the data from the onboard computer (ATmega 128), using the XBee datalink, to MATLAB version 7.6.034 installed on a laptop computer (Intel core 2 duo 1.7 GHz, 4 GB ram) running windows XP. We were able to run GNC loops at an update rate of about 0.05 seconds (20 Hz). Operation remained satisfactory at this update rate, with increase in frame drops if background processes were running on XP and as the computational complexity of the GNC algorithm increased.

VII. Flight Test Results

In the final configuration, all GNC software was programmed using the C programming language on the onboard ATmega128 microcontroller serving as the onboard computer. The microcontroller read the data from the onboard sensors, performed the required GNC computations, and executed the commands by manipulating the servos using Pulse Width Modulation (PWM) signals. A 20 Hz update rate for the controller was found to be sufficient for our purpose. The microcontroller also communicated with an off-

[g]MATHWORKS MATLAB® version 7.6.034 R2008A documentation

board ground station through the Xbee modem at an update rate of 10 Hz. An independent 900 MHz datalink was used to transmit live video information from the onboard camera to an independent ground station. The range data and vehicle status transmitted to the ground station using the Xbee modem was used for monitoring vehicle status and post-flight analysis. The Ground Control Station (GCS) was developed based on our existing GCS for outdoor UAS.[26] The GCS was programmed in C++ and used Open GL for graphical rendering. The GCS enabled real-time monitoring of all sensor data, real time adjustment of control loop gains, and functioned as a pilot interface during manual flight. A useful feature of the GCS is its ability to plot all sensor data in real time, and the ability to save the underlying sequences as MATLAB or text data files for data analysis. A screen shot of the GCS during an autonomous flight is shown in Figure 18.

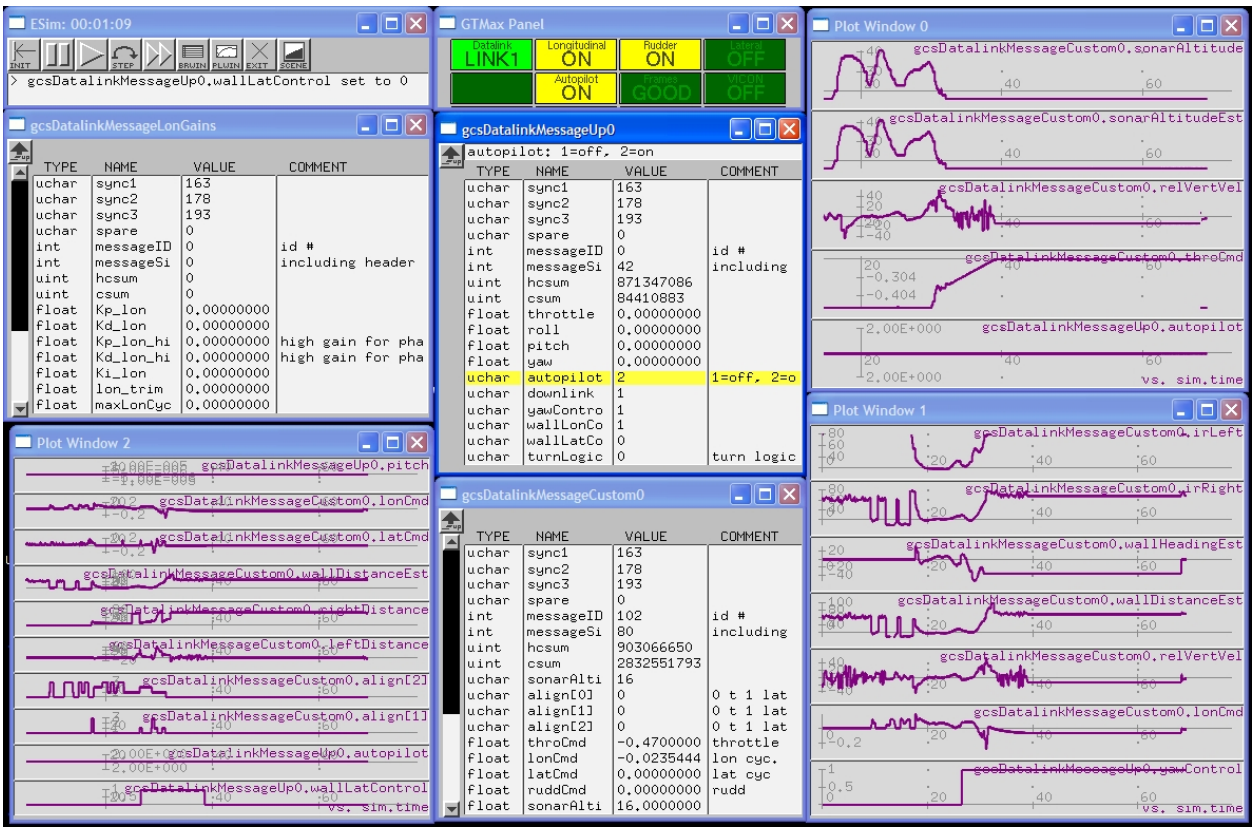


Figure 18: Screen shot of ground control station.

To date (April 2010), the aircraft has flown over 15 flights in fully autonomous mode and well over 100 flights where individual capabilities were tested. A typical fully autonomous flight test began with manual takeoff and flight to a desired altitude, altitude control loop was then enabled, followed by enabling of all other loops and the guidance logic. Flight times of approximately seven minutes were achieved with 11.1 volt Lithium Polymer battery packs, which were the only source of onboard power. Figure 19 shows the performance of the altitude control loop as the vehicle maneuvers an indoor arena at the fifth International Aerial Robotics Competition (IARC) held by the Association for Unmanned Vehicles International (AUVSI). It can be seen that the GNC algorithm is successful in separating sensor outliers from real discontinuities in the arena floor. The sonar based PID altitude control loop showed excellent performance. The performance of the wall following control loops for another flight is shown in Figure 20. In that figure, the subplot on the top shows the estimated distance to the wall as the rotorcraft maneuvers an indoor arena. The third subplot shows the distance to a wall that is in the path of the rotorcraft as the rotorcraft traverses to the right of the wall when facing a wall. The second subplot shows the longitudinal cyclic commands generated by the onboard control algorithm which attempts to keep the rotorcraft between 0.5 m and 1.52 m (20 and 60 inches) of the wall as the rotorcraft maneuvers alongside wall, and performs both inside and outside turns (discussed in Section IV). The fourth subplot shows the lateral cyclic commands generated by the onboard lateral position control algorithm which engages when a wall or an obstacle is detected. Note

that the x axis of the figure plots time since the onboard controller has been switched on and starts from around 1750 seconds for the section of the flight shown.

The vehicle was successfully able to navigate a variety of indoor geometries autonomously. These included long and angled corridors, sharp corners, flight over obstacles, and flight through windows. As the system has no access to absolute position information, it is possible to imagine situations where the presented control architecture can be suboptimal. For example, the system would have no way of telling whether it has encountered a corner, a window, or a door when the outside turn mode is engaged. This could lead to chance situations where the system enters a building through one window and leaves through another. A way to mitigate such situations is to add randomness to system behavior so that the probability of the system being stuck in repetitive loops is minimized. We found the system performance acceptable given that no absolute position information was required, and no elaborate mapping of the environment was performed. We note that the inclusion of gyroscopic rate dampers in the roll and pitch channels has the potential to improve the performance.

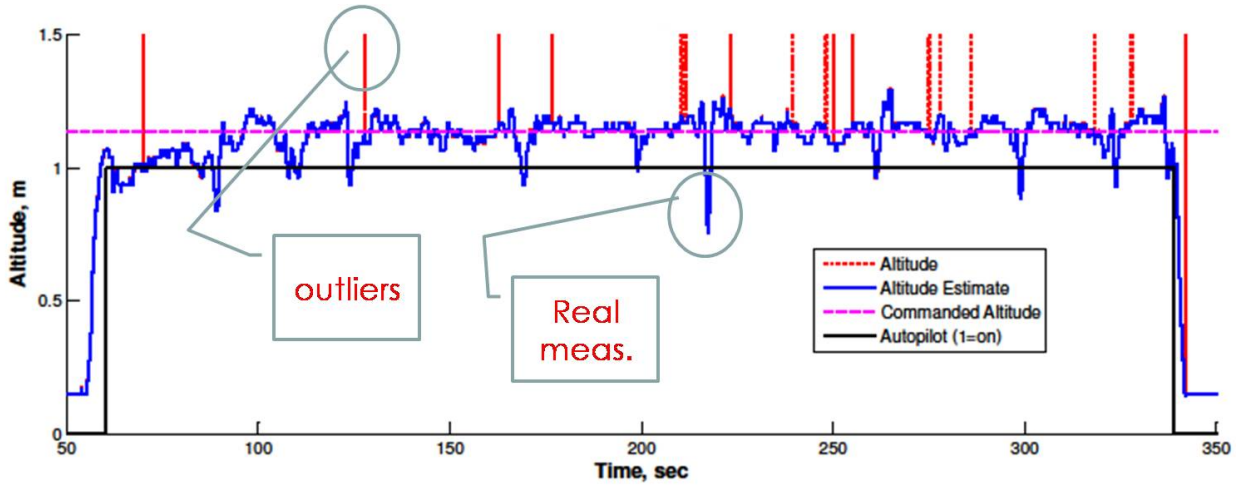


Figure 19: Altitude measurements and estimate. Note that the outlier detector filters out spikes in the measured altitude.

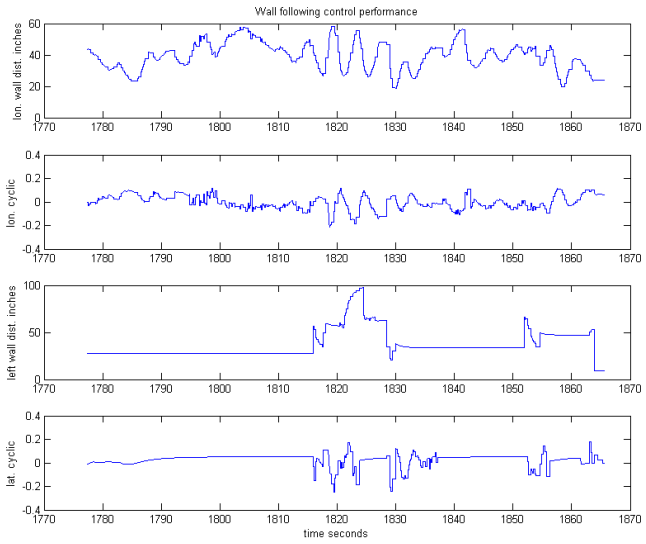


Figure 20: Wall following control loop performance, shown are longitudinal and lateral loops

VIII. Conclusion

In this paper we discussed the design, development, and flight testing effort for the Georgia Tech GT Lama rotorcraft UAS, which is a fully autonomous UAS capable of exploring cluttered indoor environments without any external sensing aids. We chose an off the shelf coaxial rotorcraft based design with desirable stick free properties in order to remain low-weight and have sufficient maneuverability. Leveraging the stick free properties of the rotorcraft platform, we developed an extremely low-cost Guidance Navigation and Control based entirely on 5 commercially available range sensors. The key innovations required in realizing this were the development of a filtering algorithm capable of detecting and rejecting sensor induced outliers, and the development of elaborate event based guidance algorithm that uses the inherent structure of the indoor environment. The guidance algorithm detected and followed walls in an indoor environment to ensure that maximum amount of indoor perimeter is explored. The algorithm did not require elaborate mapping and localization techniques, and hence was found suitable for online implementation on low-cost, low-power, off-the-shelf microcontrollers. We also discussed the details of a MATLAB based architecture that allows rapid validation of GNC algorithms on real flight hardware. We noted that this architecture is easy to realize, and greatly aids in analyzing the feasibility of GNC algorithms. Finally, we presented flight test results demonstrating the capability of the extremely low- cost(USD 900), low-weight (600g) GT Lama rotorcraft UAS and the onboard GNC algorithm to explore cluttered and confined indoor environments using only 5 range sensors. These results demonstrate that it is possible to develop fully autonomous indoor UAS without relying on elaborate sensors for full state estimation and for forming a detailed map of the environment. However, it is clear that augmenting the presented method with better sensors and mapping tools will significantly improve the performance, this requires increased payload capability and power. The requirement on extra payload and sensory power on a single aircraft may be bypassed by distributing the sensing load over a fleet of indoor vehicles.

Acknowledgments

The authors thank the members of Georgia Tech Aerial Robotics (GTAR) team: Claus Christmann, Scott Kimbrell, Syed Shah, Allen Wu, Shusaku Yamaura, and many others. The authors thank Jeong Hur for help and advice during construction of the vehicle.

References

[1]Kayton, M. and Fried, W. R., *Avionics Navigation Systems*, John Wiley and Sons, 1997.

[2]Christophersen, H. B., Pickell, W. R., Neidoefer, J. C., Koller, A. A., Kannan, S. K., and Johnson, E. N., "A compact Guidance, Navigation, and Control System for Unmanned Aerial Vehicles," *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 3, May 2006.

[3]Wendel, J., Maier, A., Metzger, J., and Trommer, G. F., "Comparison of Extended and Sigma-Point Kalman Filters for Tightly Coupled GPS/INS Integration," *AIAA Guidance Navigation and Control Conference*, San Francisco, CA, 2005.

[4]Guenard, N., Hamel, T., and Moreau, V., "Dynamic Modeling and Intuitive Control Strategy for an "X4-Flyer"," *International Conference on Control and Automation*, 2005, pp. 141–146.

[5]Martin, P. and Salaün, E., "The True Role of Accelerometer Feedback in Quadrotor Control," *IEEE International Conference on Robotics and Automation*, 2010, pp. 1623–1629.

[6]Grzonka, S., Grisetti, G., and Burgard, W., "Towards a Navigation System for Autonomous Indoor Flying," *IEEE International Conference on Robotics and Automation*, 2009.

[7]Bouabdallah, S. and Siegwart, R., "Full Control of a Quadrotor," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 153–158.

[8]Guenard, N., Hamel, T., and Mahony, R., "A Practical Visual Servo Control for an Unmanned Aerial Vehicle," *IEEE Transactions on Robotics*, Vol. 24, No. 2, 2008, pp. 331–340.

[9]Achtelik, M., Bachrach, A., He, R., Prentice, S., and Roy, N., "Stereo Vision and Laser Odometry for Autonomous Helicopters in GPS-Denied Indoor Environments," *Proceedings of SPIE*, Vol. 7332, 2009.

[10]Chowdhary, G., Salaün, E., Ottander, J., and Johnson, E., "Low Cost Guidance, Navigation, and Control Solutions for Miniature Air Vehicle in GPS Denied Environments," *First Symposium on Indoor Flight, International Aerial Robotics Competition*, Mayaguez, Puerto Rico, 2009.

[11]Chowdhary, G. et al., "Georgia Tech Aerial Robotics Team 2009 International Aerial Robotics Competition Entry," *Proc. of the 1st Symposium on Indoor Flight, International Aerial Robotics Competition*, 2009.

[12]Sobers, M. D., Chowdhary, G., and Johnson, E., "Indoor Navigation for Unmanned Aerial Vehicles," *AIAA Guidance Navigation and Control Conference*, Chicagi, IL, 2009.

[13]Gelb, A., *Applied Optimal Estimation*, MIT Press, 1974.

[14]Johnson, E. N. and Kannan, S. K., "Adaptive Trajectory Control for Autonomous Helicopters," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, May-June 2005, To appear.

[15]Wu, A. D. and Johnson, E. N., "Methods for Localization and Mapping Using Vision and Inertial Sensors," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2008.

[16]Jonghyuk, K. and Salah, S., "Autonomous Airborne Navigation in Unknown Terrain Environments," *IEEE transactions on aerospace and electronic systems*, Vol. 40, No. 3, 2004, pp. 1031.

[17]Chen, Z., Samarabandu, J., and Rodrigo, R., "Recent Advances in Simultaneous Localization and Map-building using Computer Vision," *Advanced Robotics*, Vol. 21, 2007, pp. 233–265.

[18]Kannan, S. K., *Adaptive Control of Systems in Cascade with Saturation*, Ph.D. thesis, Georgia Institute of Technology, Atlanta Ga, 2005.

[19]Chowdhary, G. and Lorenz, S., "Control of a VTOL UAV via Online Paramter Identification," *AIAA Guidance Navigation and Control Conference*, San Francisco,CA, 2005.

[20]Shim, H., Koo, T. J., Hoffman, F., and Sastry, S., "A Comprehensive Study of Control Design for an Autonomous Helicopter," *IEEE Conference on Decision and Control*, 1999.

[21]Mettler, B., *Modeling Identification and Characteristics of Miniature Rotorcrafts*, Kluwer Academic Publishers, USA, 2003.

[22]Chowdhary, G. and Lorenz, S., "Non-Linear Model Identification for a Miniature Rotorcraft, Preliminary Results," *American Helciopter Society 61st Annual Forum*, 05.

[23]Valenti, M., Bethke, B., Fiore, G., How, J., and Feron, E., "Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[24]How, J., Bethke, B., Frank, A., Dale, D., and Vian, J., "Real-Time Indoor Autonomous Vehicle Test Environment," *IEEE control systems*, Vol. 28, No. 2, 2008, pp. 51–64.

[25]Saad, E., Vian, J., Clark, G., and Bieniawski, S., "Vehicle Swarm Rapid Prototyping Testbed," *Proc. of the AIAA Infotech@Aerospace Conference*, 2009.

[26]Kannan, S. K., Koller, A. A., and Johnson, E. N., "Simulation and Development Environment for Multiple Heterogeneous UAVs," *AIAA Modeling and Simulation Technology Conference*, No. AIAA-2004-5041, Providence, Rhode Island, August 2004.